CIS Algorithm and Applications of Data Mining

Assignment 2

**1. Overview of the Assignment**

In this assignment, you will implement **Apriori algorithm** using Apache Spark Framework. You will develop a program to find frequent itemset in two datasets, one simulated dataset and one real-world dataset generated from Yelp dataset. The goal of this assignment is to apply the algorithms you have learned in class on large datasets more efficiently in a distributed environment.

**2. Requirements**

You must use Python 3.6+ and Spark 3.0+ to implement all tasks.

You need to turn in **two Python scripts**, named (all lowercase): **apriori.py, preprocess.py**

**3. Datasets**

You will use one simulated dataset and one real-world dataset.

For the simulated dataset, you will build and test your Aprioriprogram with small.csv, which is mainly used for your debugging period.

For the real-word dataset, you will first generate a subset using business.json and review.json from the Yelp dataset (https://drive.google.com/drive/folders/1q6A3s3MqqYmhkrEQQ\_sq9JTj3CH27MKc) with **the same CSV structure as the simulated data,** i.e., the first column is user\_id and the second column is business\_id. Then test your code with this real-world data.

**4. Tasks**

In this assignment, you will implement the **Apriori algorithm** on top of Apache Spark Framework. You need to find **all the frequent itemset** in any given input file.

**4.1 Task 1: Simulated data**

In task1, your program will be tested with two kinds of market-basket model.

**Case 1**: You will calculate the combinations of **frequent businesses** (as singletons, pairs, triples, etc.) that are qualified as frequent given a support threshold. That is to say, you need to create a basket for each user containing the business ids reviewed by this user. If a business was reviewed more than once by a reviewer, we consider this product was rated only once. More specifically, the business ids within each basket are unique. The generated baskets are similar to:

**user1: [business11, business12, business13, ...]**

**user2: [business21, business22, business23, ...]**

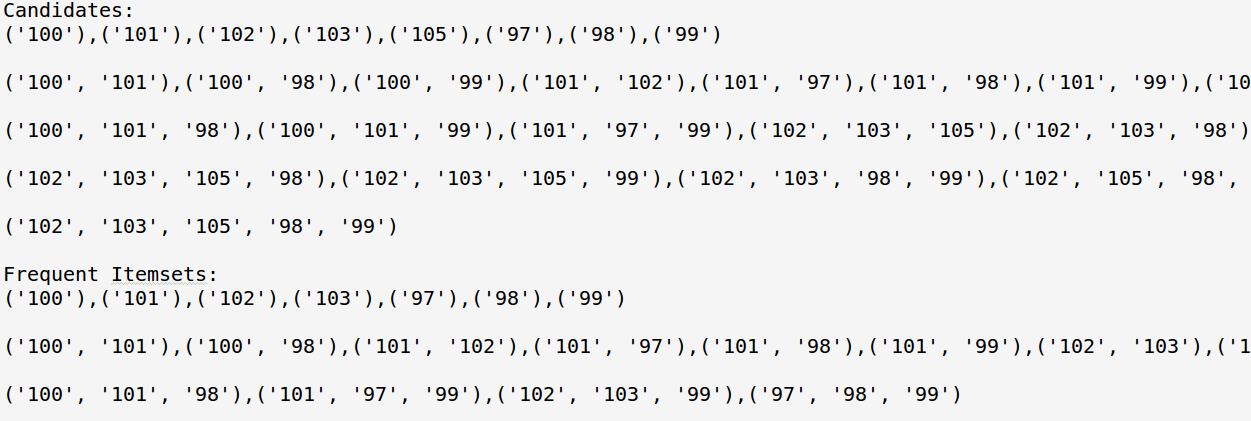
**Case 2**: You will calculate the combinations of **frequent users** (as singletons, pairs, triples, etc.) that are qualified as frequent given a support threshold. You need to create a basket for each business containing the user ids that commented on this business. Similar to case 1, the user ids within each basket are unique. The generated baskets are similar to:

**business1: [user11, user12, user13, ...]**

**business2: [user21, user22, user23, ...]**

**Output File Format:**

##### For each line, you should output the final frequent itemset for every itemset length. The printed itemset must be sorted in **lexicographical** order. Here is an example of the output file:



**Execution Command:**

python apriori.py --case <Case number> --support <Support> --input\_file <Input file path> --output\_file <Output file path>

Example: python apriori.py --case 1 --support 9 --input\_file data.csv --output\_file output.txt

1. Case number: **Integer** that specifies the case. 1 for Case1 and 2 for Case2.
2. Support: **Integer** that defines the minimum count to qualify as a frequent itemset.
3. Input file path: This is the path to the input file including path, file name and extension.
4. Output file path: This is the path to the output file including path, file name and extension.

**4.2 Task 2: Yelp data**

In task2, you will explore the Yelp dataset to find the frequent business sets (**only case 1**). You will jointly use the business.json and review.json to generate the input user-business CSV file yourselves.

**(1) Data preprocessing**

You need to generate a sample dataset from business.json and review.json with following steps:

1. The state of the business you need is Nevada, i.e., filtering “state” == “NV”
2. Select user\_id and business\_id from review.json whose business\_id is from Nevada. Each line in the CSV file would be one pair of (user\_id1, business\_id1)
3. The header of CSV file should be “user\_id,business\_id”

You need to save the dataset in CSV format. The following figure shows an example of the output file

Text, table

Description automatically generated

**(2) Apply Apriori algorithm**

The requirements for task 2 are similar to task 1. You will test your implementation with the large dataset you just generated. You are asked to find the frequent business sets (**only case 1**) from the file you just generated. The following are the steps you need to do:

1. Read the newly created CSV file and then build the case 1 market-basket model

2. Find out qualified users who reviewed more than *k* businesses. (*k* is the filter threshold)

3. Apply the Apriori algorithm to find all frequent itemset

**Output format:**

The output file format is the same with task 1.

**Execution Command:**

python apriori.py --thre < Filter threshold > --support <Support> --input\_file <Input file path> --output\_file <Output file path>

Example: python apriori.py --thre 70 --support 50 --input\_file data.csv --output\_file output.txt

1. Filter threshold: **Integer** that isused to filter out qualified users
2. Support: **Integer** that defines the minimum count to qualify as a frequent itemset.
3. Input file path: This is the path to the input file including path, file name and extension.
4. Output file path: This is the path to the output file including path, file name and extension.